Cross-validatory framework for optimal parameter estimation of KPCA and KPLS models

Fu, Y., Kruger, U., Li, Z., Xie, L., Thompson, J., Rooney, D., Hahn, J., & Yang, H. (2017). Cross-validatory framework for optimal parameter estimation of KPCA and KPLS models. *Chemometrics and Intelligent Laboratory Systems*, *167*, 196-207. https://doi.org/10.1016/j.chemolab.2017.06.007

Published in:
Chemometrics and Intelligent Laboratory Systems

Document Version:
Peer reviewed version

Cross-validatory framework for optimal parameter estimation of KPCA and KPLS models

Yujia Fu[a,b], Uwe Kruger[a,*], Zhe Li[c], Lei Xie[d], Jillian Thompson[e], David Rooney[e], Juergen Hahn[a], Huizhong Yang[b,*]

[a]*Department of Biomedical Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.*
[b]*Key Laboratory of Advanced Process Control for Light Industry, Jiangnan University, Wuxi 214122, Jiangsu, P.R. China.*
[c]*School of Hydraulic, Energy and Power Engineering, Yangzhou University, Yangzhou, 225127, P.R. China*
[d]*State Key Lab of Industrial Control Technology, Zhejiang University, Hangzhou 310027, P.R. China*
[e]*School of Chemistry and Chemical Engineering, Queen's University Belfast, BT9 5AG, U.K.*

Abstract

This article revisits recently proposed methods to determine the kernel parameter and the number of latent components for identifying kernel principal component analysis (KPCA) and kernel partial least squares (KPLS) models. A detailed analysis shows that existing work is neither optimal nor efficient in determining these important parameters and may lead to erroneous estimates. In addition to that, most methods are not designed to simultaneously estimate both parameters, *i.e.* they require one parameter to be predetermined. To address these practically important issues, the article introduces a cross-validatory framework to optimally determine both parameters. Application studies to a simulation example and a total of three experimental or industrial data sets confirm that the cross-validatory framework outperforms existing methods and yields optimal estimations for both parameters. In sharp contrast, existing work has the potential to substantially overestimate the number of latent components and to provide inadequate estimates for the kernel parameter.

Keywords: Nonlinear models, cross-validatory framework, optimal parameter estimation, kernel parameter, number of latent variable sets, combined objective function

1. Introduction

In chemometrics, identifying accurate latent variable models is of fundamental importance (i) for extracting information from spectra, *e.g.* obtained by QSAR [1], Raman [2] or IR [3] spectroscopy, and (ii) for applications to process systems engineering [4], petrochemical processes [5] and process monitoring [6, 7, 8]. The first thrust in research focused on applications of principal component analysis (PCA) and partial least squares (PLS) [9, 10, 11], which assume linear relationships between the latent and recorded variable sets. This necessitated the development of their nonlinear counterparts over the past few decades [12, 13, 14, 15]. Kruger *et al.* [16] showed that KPCA [17, 18] is a generic nonlinear extension of PCA. Generic kernel-based extensions for PLS, termed KPLS, have also been developed [19, 20].

Establishing KPCA and KPLS models based on a random vector $\boldsymbol{z}$, and random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, requires the estimation of the correct number of latent variable sets and the optimal value of kernel parameters. Without restriction of generality, we assume here that the random vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ are of dimension $(N_x \times 1)$, $(N_y \times 1)$ and $(N_z \times 1)$, respectively, have unknown distributions with zero mean vectors and bounded covariance matrices. This article develops algorithms to *optimally determine the number of latent sets, n, and the unknown kernel parameter, σ,* by utilizing *objective functions that contain both parameters.* The next three subsections summarize existing work on how to estimate n and σ to highlight that the literature has not proposed such an optimal framework.

*Corresponding authors

Email addresses: fyj@vip.jiangnan.edu.cn (Yujia Fu), krugeu@rpi.edu (Uwe Kruger), lizhe@yzu.edu.cn (Zhe Li), leix@iipc.zju.edu.cn (Lei Xie), jillian.thompson@qub.ac.uk (Jillian Thompson), d.rooney@qub.ac.uk (David Rooney), hahnj@rpi.edu (Juergen Hahn), yhz@jiangnan.edu.cn (Huizhong Yang)

1.1. Estimating n and σ for KPCA

To estimate n, the cumulative percent variance (CPV) [21, 22], measuring the variance captured by the first few PCs, has been proposed. This technique, however, is prone to be subjective. Another technique is kernel parallel analysis (KPA) [23], which is an automated technique and an extension of parallel analysis for PCA [24]. More recently, a technique that is based on the reconstruction error was proposed [25], which determines n such that the residual error is sufficiently small, which is also subjective. Ref [26] proposed an approach that relies on dividing the data set into 5 segments and determines n using cross-validation. It should be noted that the methods in Refs [23, 25, 26] require an *a priori* selection of σ .

To determine σ, Teixeira *et al.* [27] suggested that it is a function of the maximum distance of the samples from the sample mean vector. This approach is termed here the maximum distance to mean or MDM technique. Ni *et al.* [28] selected σ as the sum of the difference of the maximum and minimum for each variable in the data matrix, referred to here as the sum of the variable spread or SVS approach. More recently, Kenig *et al.* [29] suggested using the 0.2 quantile of the distances between the samples as a guide to select the kernel parameter, defined here as the sample distance or SD criteria. Finally, Deng and Tian [30] proposed determining σ as 100 times N_x, referred to here as the PVC technique.

1.2. Estimating n and σ for KPLS

To estimate n, the literature advocated a 5-fold cross-validatory approach [31, 32, 33]. This approach, however, requires predetermining σ and may not work well for small data sets. For estimating σ, a heuristic and subjective approach was proposed in Ref [20], referred to here as dimension and variance or DaV criteria, selecting σ as a product of the variable variances, the number of variables in the input space and a constant between 1 and 10. Monte Carlo cross-validation (MCCV) was proposed by Shinzawa *et al.* [34], which is based on a resampling technique that randomly selects sets of testing samples. Both, the selection of the total number of resampling and calibration sets, which must be pre-defined, however, is still an open question. Moreover, the random resampling is computationally expensive and may lead to certain samples being used more than once for assessing model performance or not being used for model identification. Finally, a simulated annealing (SA) method was proposed to automatically select σ [35]. As a global optimal solution, SA can successfully avoid local minima but it is time-consuming and requires setting initial parameters. As for KPCA, each of these methods require n to be predetermined.

1.3. Other more general methods for estimating σ

Kernel target alignment (KTA) [36] is a kernel matrix evaluation criteria used to measure the degree of linear dependency between the kernel matrix and a target. Another technique is the feature space-based kernel matrix (FSM) evaluation measure [37]. Both KTA and FSM, however, may not work well for small sample sizes and have a tendency of overfitting [38]. Addressing these deficiencies, Yang *et al.* [38] developed a technique based on the largest variance criteria (LVC) to estimate the kernel parameter. Unlike PCA, however, a maximum variance may not guarantee that the extracted components strongly correlate to $\boldsymbol{z}$ and $\boldsymbol{y}$. A grid search strategy is widely used to optimally determine σ. However, it may yield a local minima if the range is selected to be too small [34, 35]. To examine the data distribution more closely, Zhang *et al.* [39] suggested selecting the median value of the reciprocal distances between each sample and the sample mean as the optimal σ, which is referred to here as the MID criteria. Other research streams are more closely related to performance measures and include the distance of reference samples to the furthest and nearest neighbors (DFN) [40], and include intelligent optimization, *e.g.* genetic algorithms (GA) [41, 42]. Each method, however, requires a preestimate of n.

1.4. Motivation for this work

Following from the preceding discussion and based on the application of existing work in Sections 4 and 5, the literature has not proposed an *optimal estimation of both parameters* that is based on an *objective function that includes both parameters*, necessitating the work in this article. The core contribution here is the development of an optimal cross-validatory framework for identifying KPCA and KPLS models, based on an objective function that relies on the average prediction error for samples removed (KPCA/KPLS) and variables removed (KPCA). The optimal parameter set results in the best model prediction, *i.e.* a minimum of the objective function. It should be noted that the use of the model prediction error is in line with the properties of KPCA/KPLS, which Subsection 5.4 further elaborates upon.

1.5. Organization of this article

A brief summary of KPCA and KPLS is given in the next section. Then, Section 3 introduces the cross-validatory framework for KPCA and KPLS to optimally estimate n and σ. Sections 4 and 5 compare the performance of this framework to existing work on the basis of a simulation example and three industrial or experimental data sets, respectively. Finally, a concluding summary is given in Section 6.

2. Preliminaries

This section provides a brief summary of KPCA and KPLS in Subsections 2.1 and 2.2, respectively.

2.1. Kernel principal component analysis

KPCA first maps a set of L data points of $\boldsymbol{z}$, drawn independently, onto a high-dimensional feature space $\boldsymbol{f}$, being of dimension $M \leq \infty$, based on the nonlinear transformation $\boldsymbol{f} = \boldsymbol{\phi}(\boldsymbol{z})$, which yields $\boldsymbol{\Phi}^T = [\ \boldsymbol{\phi}(\boldsymbol{z}_1) \quad \boldsymbol{\phi}(\boldsymbol{z}_2) \quad \cdots \quad \boldsymbol{\phi}(\boldsymbol{z}_L)\ ]\ (M \times L)$. Based on Cover's theorem [43], the transformed data points fall in the vicinity of an n_f dimensional plane in the feature space. The estimated covariance matrix of $\boldsymbol{f}$ is:

$$\widehat{\boldsymbol{C}}_f = \tfrac{1}{L-1}\overline{\boldsymbol{\Phi}}^T\overline{\boldsymbol{\Phi}}\ (M \times M). \tag{1}$$

Here, $\bar{\cdot}$ and $\widehat{\cdot}$ denote a matrix which stores data points of a random vector that are mean centered and a parameter estimate, respectively. Secondly, as $\boldsymbol{\phi}(\cdot)$ is usually unknown, the KPCA model is determined based on the centered Gram matrix:

$$\boldsymbol{G}_f = \overline{\boldsymbol{\Phi}}\ \overline{\boldsymbol{\Phi}}^T\ (L \times L), \tag{2}$$

where, $\overline{\boldsymbol{\Phi}} = \boldsymbol{\Phi} - \frac{1}{L}\mathbf{1}\mathbf{1}^T\boldsymbol{\Phi}$ with $\mathbf{1}$ being a vector of L ones. Using the definition of the kernel matrix $\boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z}) = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\ (L \times L)$, $\boldsymbol{Z}^T = [\ \boldsymbol{z}_1 \quad \boldsymbol{z}_2 \quad \cdots \quad \boldsymbol{z}_L\ ]\ (N_z \times L)$, the Gram matrix is given by:

$$\boldsymbol{G}_f = \boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z}) - \tfrac{1}{L}\Big(\boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z})\mathbf{1}\Big)\mathbf{1}^T - \tfrac{1}{L}\mathbf{1}\Big(\boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z})\mathbf{1}\Big)^T + \tfrac{1}{L^2}\mathbf{1}\Big(\mathbf{1}^T\boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z})\mathbf{1}\Big)\mathbf{1}^T. \tag{3}$$

The kernel matrix stores the scalar products $K_f\big(\boldsymbol{z}_i,\boldsymbol{z}_j\big) = \boldsymbol{\phi}^T(\boldsymbol{z}_i)\ \boldsymbol{\phi}(\boldsymbol{z}_j)$, which, based on the properties of reproducing kernels, can be constructed from various kernel functions, *e.g.* the Gaussian kernel:

$$K_f\big(\boldsymbol{z}_i,\boldsymbol{z}_j\big) = \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2}{\sigma_f^2}\right). \tag{4}$$

Here, σ_f is the kernel parameter and $\|\cdot\|^2$ is the squared Euclidean distance. As the feature transformation is designed to map the L data points to be in the vicinity of a plane of dimension n_f that is embedded within the high dimensional features space, the KPCA model is given by the eigendecomposition of $\boldsymbol{G}_f$, *i.e.*:

$$\boldsymbol{G}_f\ \boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i \qquad i = \big\{1,2,\ldots,L\big\}, \tag{5}$$

where λ_i and $\boldsymbol{v}_i$ are the ith largest eigenvalue and its corresponding L-dimensional eigenvector of $\boldsymbol{G}_f$, respectively. Thirdly, the score vector $\boldsymbol{t}$ of dimension n_f for a data point of $\boldsymbol{z} \notin \boldsymbol{Z}$ can now be computed as:

$$\boldsymbol{t} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^T\overline{\boldsymbol{\Phi}}\ \overline{\boldsymbol{\phi}}^T(\boldsymbol{z}) = \underbrace{\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^T\left[\boldsymbol{I} - \tfrac{1}{L}\mathbf{1}\mathbf{1}^T\right]}_{\boldsymbol{A}^T}\Big(\boldsymbol{k}_f(\boldsymbol{Z},\boldsymbol{z}) - \underbrace{\tfrac{1}{L}\boldsymbol{K}_f(\boldsymbol{Z},\boldsymbol{Z})\mathbf{1}}_{\overline{\boldsymbol{k}}_f}\Big) = \boldsymbol{A}^T\Big(\boldsymbol{k}_f(\boldsymbol{Z},\boldsymbol{z}) - \overline{\boldsymbol{k}}_f\Big), \tag{6}$$

with $\boldsymbol{\Lambda}$ and $\boldsymbol{V}$ storing the n_f largest eigenvalues and corresponding eigenvectors, respectively, $\boldsymbol{I}$ being the L dimensional identify matrix and $\boldsymbol{k}_f^T(\boldsymbol{Z},\boldsymbol{z}) = \big(\ K_f(\boldsymbol{z}_1,\boldsymbol{z}) \quad \cdots \quad K_f(\boldsymbol{z}_L,\boldsymbol{z})\ \big)$.

2.2. Kernel partial least squares

Kernel partial least square is based on the standard PLS algorithm [44]. As for KPCA, KPLS relies on a nonlinear transformation of the variable set $\boldsymbol{x}$, referred to here as the predictor variable set, *i.e.* $\boldsymbol{h} = \boldsymbol{\psi}(\boldsymbol{x})$ [19, 20]. In a similar fashion to KPCA, scalar products of the feature vectors $\boldsymbol{\psi}^T(\boldsymbol{x}_i)\ \boldsymbol{\psi}(\boldsymbol{x}_j)$ can be described by kernel functions $K_h\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) = \exp\big(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma_h^2\big)$, where σ_h is the kernel parameter. The basic PLS algorithm relies on iteratively computing the eigenvectors of the matrix product $\boldsymbol{Y}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{Y}$ [6], where $\boldsymbol{X}^T = [\ \boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_L\ ]$ $(N_x \times L)$ and $\boldsymbol{Y}^T = [\ \boldsymbol{y}_1 \quad \boldsymbol{y}_2 \quad \cdots \quad \boldsymbol{y}_L\ ]$ $(N_y \times L)$ are matrices storing data points of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. For KPLS, this product becomes $\boldsymbol{Y}^T\boldsymbol{G}_h\boldsymbol{Y}$, with $\boldsymbol{G}_h$ being the Gram matrix of $\boldsymbol{x}$, defined analogously to $\boldsymbol{G}_f$ in Eq. (3). Algorithm 1 summarizes the steps of the basic KPLS algorithm [20], where ${}_k\boldsymbol{t}$ and ${}_k\boldsymbol{u}$, both being L dimensional, $1 \leq k \leq n_h$, are the n_h score vectors of the L data points of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and ${}_k\boldsymbol{c}$ is a loading vector of $\boldsymbol{y}$. Similar to PLS, KPLS estimates a regression matrix

Setup ${}_1\boldsymbol{Y} = \boldsymbol{Y}$ and ${}_1\boldsymbol{G}_h$;
for $k = 1 : n_h$ **do**
  Randomly initialize ${}_k\boldsymbol{u}^{(0)}$;
  Set $j = 0$ and $e = 100$;
  while $e < 1e - 10$ **do**
    ${}_k\boldsymbol{t}^{(j)} = {}_k\boldsymbol{G}_h\ {}_k\boldsymbol{u}^{(j)}$;
    ${}_k\boldsymbol{t}^{(j)} = {}_k\boldsymbol{t}^{(j)}/\|{}_k\boldsymbol{t}^{(j)}\|$;
    ${}_k\boldsymbol{c}^{(j)} = {}_k\boldsymbol{Y}^T\,{}_k\boldsymbol{t}^{(j)}$;
    ${}_k\boldsymbol{u}^{(j)} = {}_k\boldsymbol{Y}\ {}_k\boldsymbol{c}^{(j)}$;
    ${}_k\boldsymbol{u}^{(j+1)} = {}_k\boldsymbol{u}^{(j)}/\|{}_k\boldsymbol{u}^{(j)}\|$;
    $e = \|{}_k\boldsymbol{u}^{(j+1)} - {}_k\boldsymbol{u}^{(j)}\|$;
    $j = j + 1$;
  end
  ${}_{k+1}\boldsymbol{G}_h = \big[\boldsymbol{I} - {}_k\boldsymbol{t}\ {}_k\boldsymbol{t}^T\big]_k\boldsymbol{G}_h\big[\boldsymbol{I} - {}_k\boldsymbol{t}\ {}_k\boldsymbol{t}^T\big]$;
  ${}_{k+1}\boldsymbol{Y} = \big[\boldsymbol{I} - {}_k\boldsymbol{t}\ {}_k\boldsymbol{t}^T\big]_k\boldsymbol{Y}$;
end

Algorithm 1: KPLS algorithm

$\boldsymbol{B}$ for predicting $\boldsymbol{y}$, *i.e.* $\boldsymbol{y}^T = \overline{\boldsymbol{\psi}}^T(\boldsymbol{x})\ \boldsymbol{B} + \boldsymbol{e}^T = \widehat{\boldsymbol{y}}^T + \boldsymbol{e}^T$, where the vector $\boldsymbol{e}$ represents the modeling error:

$$\boldsymbol{B} = \overline{\boldsymbol{\Psi}}^T\boldsymbol{U}\big[\boldsymbol{T}^T\boldsymbol{G}_h\boldsymbol{U}\big]^{-1}\boldsymbol{T}^T\boldsymbol{Y}, \tag{7}$$

where $\boldsymbol{T} = [\ {}_1\boldsymbol{t} \quad {}_2\boldsymbol{t} \quad \cdots \quad {}_{n_h}\boldsymbol{t}\ ]$ and $\boldsymbol{U} = [\ {}_1\boldsymbol{u} \quad {}_2\boldsymbol{u} \quad \cdots \quad {}_{n_h}\boldsymbol{u}\ ]$, with both matrices being of the dimension $(L \times n_h)$. As the nonlinear transformation $\boldsymbol{\psi}(\cdot)$ is not known, the prediction of a data point of $\boldsymbol{y}$ using its corresponding data point of $\boldsymbol{x} \notin \mathbf{X}$ is given by:

$$\widehat{\boldsymbol{y}}^T = \overline{\boldsymbol{\psi}}^T(\boldsymbol{x})\ \boldsymbol{B} = \overline{\boldsymbol{\psi}}^T(\boldsymbol{x})\ \overline{\boldsymbol{\Psi}}^T\boldsymbol{U}\big[\boldsymbol{T}^T\boldsymbol{G}_h\boldsymbol{U}\big]^{-1}\boldsymbol{T}^T\boldsymbol{Y}. \tag{8}$$

Similar to Eq (6), the L-dimensional vector $\overline{\boldsymbol{\Psi}}\ \overline{\boldsymbol{\psi}}(\boldsymbol{x})$ contains the mean centered scalar products $K_h(\boldsymbol{x}_i, \boldsymbol{x})$, $1 \leq i \leq L$.

3. Cross-validatory framework for jointly estimating n and σ

Subsections 3.1 and 3.2 define the objective functions and the algorithms to optimally estimate n and σ for KPCA and KPLS, respectively.

3.1. Objective function and algorithm for KPCA

The subsection first revises the recently introduced two-dimensional cross-validatory algorithm [45] to optimally estimate n_f. As this algorithm, which can be seen as a nonlinear extension of the work in Refs [46, 47], does not yield an estimation of σ_f, the subsection then introduces an algorithm for optimally estimating σ_f when n_f is known. This is the first contribution of this article.

3.1.1. Algorithm for optimally estimating n_f [45]

Following the discussion in Refs [17, 45], kernel principal component regression (KPCR) can be applied to estimate the reconstruction, or inverse, mapping from $\boldsymbol{f}$ to $\boldsymbol{z}$. This gives rise to the following two-dimensional cross-validatory approach, which relies on segmenting $\boldsymbol{Z}$ as follows [45]:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{\xi}_1^{(1)} & \cdots & \boldsymbol{\xi}_i^{(1)} & \cdots & \boldsymbol{\xi}_{N_z}^{(1)} \\ \vdots & & \vdots & & \vdots \\ \boldsymbol{\xi}_1^{(j)} & \cdots & \boldsymbol{\xi}_i^{(j)} & \cdots & \boldsymbol{\xi}_{N_z}^{(j)} \\ \vdots & & \vdots & & \vdots \\ \boldsymbol{\xi}_1^{(m)} & \cdots & \boldsymbol{\xi}_i^{(m)} & \cdots & \boldsymbol{\xi}_{N_z}^{(m)} \end{bmatrix}, \tag{9}$$

We first remove the observations of the ith variable ($1 \leq i \leq N_z$) and store them in the vector $\boldsymbol{\xi}_i$. The observations of the remaining variables are stored in the matrix $\boldsymbol{\Xi}_{-i}$. This is the first cross-validatory dimension. Next, we divide $\boldsymbol{\xi}_i$ and $\boldsymbol{\Xi}_{-i}$ further into $\boldsymbol{\xi}_i^{(j)}$ and $\boldsymbol{\xi}_i^{(-j)}$ (omitting $\boldsymbol{\xi}_i^{(j)}$), and $\boldsymbol{\Xi}_{-i}^{(j)}$ and $\boldsymbol{\Xi}_{-i}^{(-j)}$ (omitting $\boldsymbol{\Xi}_{-i}^{(j)}$), respectively. This constitutes the second cross-validatory dimension. The dimensions of $\boldsymbol{\xi}_i$, $\boldsymbol{\Xi}_{-i}$, $\boldsymbol{\xi}_i^{(j)}$, $\boldsymbol{\xi}_i^{(-j)}$, $\boldsymbol{\Xi}_{-i}^{(j)}$ and $\boldsymbol{\Xi}_{-i}^{(-j)}$ are $L \times 1$, $L \times (N_z - 1)$, $p \times 1$, $(L-p) \times 1$, $p \times (N_z - 1)$ and $(L-p) \times (N_z - 1)$, respectively. For simplicity, we assume here that $p = L/m$, with m being the number of segments, has no division remainder. Under this assumption, each segment, *i.e.* $\boldsymbol{\xi}_i^{(j)}$, $j = 1, \ldots, m$ and $i = 1, \ldots, N_z$, has the same number of observations.

Utilizing $\boldsymbol{\xi}_i^{(-j)}$ and $\boldsymbol{\Xi}_{-i}^{(-j)}$, we can establish the following regression equation in the feature space:

$$\boldsymbol{\xi}_i^{(-j)} = \overline{\boldsymbol{\Phi}}_{-i}^{(-j)} \boldsymbol{\beta}_{-i}^{(-j)} + \boldsymbol{\varepsilon}_i^{(-j)}, \tag{10}$$

where $\overline{\boldsymbol{\Phi}}_{-i}^{(-j)}$ is the mean centered feature matrix of $\boldsymbol{\Xi}_{-i}^{(-j)}$, $\boldsymbol{\beta}_{-i}^{(-j)}$ ($(L-p) \times 1$) is the KPCR regression vector and $\boldsymbol{\varepsilon}_i^{(-j)}$ is an error vector [45]. As the specific form of the nonlinear function $\boldsymbol{\phi}(\cdot)$ is usually unknown, KPCR utilizes the Gram matrix of $\boldsymbol{\Xi}_{-i}^{(-j)}$ and retains $1 \leq \tilde{n}_f \leq n_{f_{max}}$, $n_{f_{max}} \leq L - p$, latent variable sets to estimate $\boldsymbol{\beta}_{-i}^{(-j)}$ [45]. To have a statistically independent assessment of the model performance, the regression model is applied to the observations stored in $\boldsymbol{\xi}_i^{(j)}$ and $\boldsymbol{\Xi}_{-i}^{(j)}$:

$$\boldsymbol{\varepsilon}_i^{(j)} = \boldsymbol{\xi}_i^{(j)} - \overline{\boldsymbol{\Phi}}_{-i}^{(j)} \widehat{\boldsymbol{\beta}}_{-i}^{(-j)} \tag{11}$$

Algorithm 1 in Ref [45] summarizes this two-dimensional cross-validatory technique, which yields a minimum of the objective function:

$$n_f = \arg\min_{\tilde{n}_f} \tfrac{1}{LN_z} \sum_{j=1}^{m} \sum_{i=1}^{N_z} \boldsymbol{\varepsilon}_i^{(j)^T}(\widetilde{n}_f)\, \boldsymbol{\varepsilon}_i^{(j)}(\widetilde{n}_f) \tag{12}$$

Although this two-dimensional cross-validatory scheme produces an optimal estimate for n_f, as a simulation example and three application studies to experimental data in Ref [45] showed, it does not provide an optimal estimate for σ_f. This follows from (i) the optimal σ_f is different for each of the $1 \leq i \leq N_z$ and $1 \leq j \leq m$ Gram matrices and (ii) that $\boldsymbol{\Xi}_{-i}$ only stores $N_z - 1$ variables and not the N_z original variables in $\boldsymbol{Z}$. Consequently, although Algorithm 1 in Ref [45] produces an optimal estimate of n_f, the optimal estimate of σ_f must be determined afterwards in a second step, which is discussed next.

3.1.2. Algorithm for optimally estimating σ_f

After Algorithm 1 in Ref [45] produced an optimal estimate of n_f, Algorithm 2 below can optimally estimate σ_f. Based on Eq (9), Algorithm 2 first defines $\boldsymbol{Z}^{(j)} = [\ \boldsymbol{\xi}_1^{(j)} \ \cdots \ \boldsymbol{\xi}_i^{(j)} \ \cdots \ \boldsymbol{\xi}_{N_z}^{(j)}\ ]$ and $\boldsymbol{Z}^{(-j)}$, which stores the remaining data points in $\boldsymbol{Z}$. Next, using the data points stored in $\boldsymbol{Z}^{(-j)}$, Algorithm 2 computes the corresponding Gram matrix. Based on Eq (5), it then determines the n_f dominant eigenpairs, *i.e.* eigenvalues and corresponding eigenvectors. This is followed by calculating the $L - p$ vectors of $\boldsymbol{t}$ using Eq (6). Finally, Algorithm 2 determines the inverse mapping using KPCR, which completes the model building stage.

Define m, $p = L/m$, set initial $\widetilde{\sigma}_f$ and set n_f (optimal estimate from Algorithm 1 in Ref [45]);
while *Check convergence* **do**
  Set $J_f(\widetilde{\sigma}_f) = 0$;
  for *j=1:m* **do**
    Define $\boldsymbol{Z}^{(-j)}$ and $\boldsymbol{Z}^{(j)}$;
    Construct $\boldsymbol{G}_f^{(-j)}$ and $\boldsymbol{G}_f^{(j)}$;
    Compute eigendecomposition of $\boldsymbol{G}_f^{(-j)}$;
    Calculate $\boldsymbol{t}^{(-j)}$ for all data points in $\boldsymbol{Z}^{(-j)}$;
    Estimate KPCR regression vector for inverse mapping;
    Calculate $\boldsymbol{t}^{(j)}$ for all data points in $\boldsymbol{Z}^{(j)}$;
    Determine prediction for $\boldsymbol{Z}^{(j)}$, $\widehat{\boldsymbol{Z}}^{(j)}$, using inverse mapping;
    Compute $\boldsymbol{E}^{(j)} = \boldsymbol{Z}^{(j)} - \widehat{\boldsymbol{Z}}^{(j)}$;
    Update $J_f(\widetilde{\sigma}_f) = J_f(\widetilde{\sigma}_f) + \|\boldsymbol{E}^{(j)}\|^2$;
  end
  Scale $J_f(\widetilde{\sigma}_f) = \frac{1}{LN_z} J_f(\widetilde{\sigma}_f)$;
  if *Converged* **then**
    End while loop;
    $\sigma_f = \widetilde{\sigma}_f$;
  else
    Update kernel parameter $\widetilde{\sigma}_f$;
  end
end

Algorithm 2: Cross-validatory method for optimally estimating σ_f — KPCA.

To independently assess the performance of the identified KPCA model, we now apply the mapping function to compute the p vectors of the random score vector $\boldsymbol{t}$ and the p reconstructed data points of $\boldsymbol{z}$, stored in $\boldsymbol{Z}^{(j)}$, using the inverse mapping of KPCR model. The cross-validatory scheme is applied until each data point of the random vector $\boldsymbol{z}$, stored in the data matrix $\boldsymbol{Z}$, has been used to independently assess the performance of the KPCA model. Defining the prediction of $\boldsymbol{z}$, using the inverse mapping, as $\widehat{\boldsymbol{z}}$ and the prediction error as $\boldsymbol{\varepsilon} = \boldsymbol{z} - \widehat{\boldsymbol{z}}$, the objective function for optimally estimating σ_f is as follows:

$$\widehat{\sigma}_f = \arg\min_{\widetilde{\sigma}_f} J_f(\widetilde{\sigma}_f) = \arg\min_{\widetilde{\sigma}_f} \frac{1}{LN_z} \sum_{j=1}^{m} {\boldsymbol{\varepsilon}^{(j)}}^T(\widetilde{\sigma}_f)\, \boldsymbol{\varepsilon}^{(j)}(\widetilde{\sigma}_f) \tag{13}$$

The design of Algorithm 2 is tailored to the application of an iterative optimization method, *e.g.* a particle swarm, gradient-based or genetic algorithm optimizer. The convergence criterion can be defined with respect to the specific optimization method, for example if the difference of the objective function in Eq (13) between two consecutive iterations is below a predefined threshold. For simplicity, we applied a grid search for determining the optimal value for σ_f in Sections 4 and 5, although this is computationally inferior.

3.2. Objective function and algorithm for KPLS

This subsection introduces a cross-validatory technique to simultaneously estimate n_h and σ_h, which is the second contribution of this article. In a similar fashion to Eq (9), the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are divided into a total of m ($1 < m \leq L$) disjoint segments of $p = L/m$ data points. This yields the matrices $\boldsymbol{X}^{(j)}$ ($p \times N_x$), $\boldsymbol{Y}^{(j)}$ ($p \times N_y$), $\boldsymbol{X}^{(-j)}$ ($(L-p) \times N_x$) and $\boldsymbol{Y}^{(-j)}$ ($(L-p) \times N_y$). As before, we assume, for simplicity, that L/m does not yield a division remainder. In a similar fashion to both cross-validatory algorithms in Subsection 3.1, we assess the KPLS model using the observations stored in $\boldsymbol{X}^{(j)}$ ($p \times N_x$), $\boldsymbol{Y}^{(j)}$ ($p \times N_y$) and identify the KPLS model based on the observations stored in $\boldsymbol{X}^{(-j)}$ ($(L-p) \times N_x$) and $\boldsymbol{Y}^{(-j)}$ ($(L-p) \times N_y$) for $1 \leq j \leq m$. This guarantees an independent evaluation as any $\boldsymbol{x}_\ell \in \boldsymbol{X}^{(j)}$ and $\boldsymbol{y}_\ell \in \boldsymbol{Y}^{(j)}$ are not stored in $\boldsymbol{X}^{(-j)}$ and $\boldsymbol{Y}^{(-j)}$, respectively. The identification of a KPLS model relies on Algorithm 1 by replacing ${}_1\boldsymbol{Y}$ and ${}_1\boldsymbol{G}_h$ with ${}_1\boldsymbol{Y}^{(-j)}$ and ${}_1\boldsymbol{G}_h^{(-j)}$, respectively. After identifying a KPLS model for a specific $\widetilde{n}_h$ and $\widetilde{\sigma}_h$,

Eq 8 can be used to compute the prediction of $\boldsymbol{Y}^{(j)}$, *i.e.* $\widehat{\boldsymbol{Y}}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h)$. On the basis of the prediction error, $\boldsymbol{E}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h) = \boldsymbol{Y}^{(j)} - \widehat{\boldsymbol{Y}}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h)$, $1 \leq j \leq m$ the objective function for determining optimal estimates of n_h and σ_h is as follows:

$$\begin{pmatrix} n_h & \sigma_h \end{pmatrix}^T = \arg\min_{\widetilde{n}_h, \widetilde{\sigma}_h} J_h(\widetilde{n}_h, \widetilde{\sigma}_h) = \arg\min_{\widetilde{n}_h, \widetilde{\sigma}_h} \frac{1}{LN_y} \sum_{j=1}^{m} \left\| \boldsymbol{E}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h) \right\|^2 \tag{14}$$

Algorithm 3 formalizes the sequence of all required steps and yields an *optimal* and *simultaneous* estimate of n_h and σ_h. Recall that the objective function in Eq 14 relies on evaluating the model performance using *independently* drawn data points, *i.e.* data points that are not used to identify the KPLS model.

Define m, $p = L/m$, $n_{h_{max}}$ and set initial $\widetilde{\sigma}_h$;
while *Check convergence* **do**
  for $\widetilde{n}_h = 1 : n_{max}$ **do**
    Set $J_h(\widetilde{n}_h, \widetilde{\sigma}_h) = 0$;
    for $j = 1 : m$ **do**
      Define $\boldsymbol{X}^{(j)}$, $\boldsymbol{X}^{(-j)}$, $\boldsymbol{Y}^{(j)}$ and $\boldsymbol{Y}^{(-j)}$;
      Construct $\boldsymbol{G}_h^{(j)}$ and $\boldsymbol{G}_h^{(-j)}$;
      Calculate KPLS model based on $\boldsymbol{X}^{(-j)}$ and $\boldsymbol{Y}^{(-j)}$ using Algorithm 1;
      Determine prediction for $\boldsymbol{Y}^{(j)}$, $\widehat{\boldsymbol{Y}}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h)$, using Equation 8;
      Compute $\boldsymbol{E}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h) = \boldsymbol{Y}^{(j)} - \widehat{\boldsymbol{Y}}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h)$;
      Update $J_h(\widetilde{n}_h, \widetilde{\sigma}_h) = J_h(\widetilde{n}_h, \widetilde{\sigma}_h) + \|\boldsymbol{E}^{(j)}(\widetilde{n}_h, \widetilde{\sigma}_h)\|^2$;
    end
  end
  Scale $J_h(\widetilde{n}_h, \widetilde{\sigma}_h) = \frac{1}{LN_y} J_h(\widetilde{n}_h, \widetilde{\sigma}_h)$;
  if *Converged* **then**
    End while loop;
    $\sigma_h = \widetilde{\sigma}_h$, $n_h = \widetilde{n}_h$;
  else
    Update kernel parameter $\widetilde{\sigma}_h$;
  end
end

Algorithm 3: Cross-validatory method for optimally and simultaneously estimating n_h and σ_h — KPLS

As before, Algorithm 3 is designed to be embedded within an iterative optimizer, *e.g.* a gradient-based, particle swarm or genetic algorithm optimizer. The results reported in this article, however, are based on a simple grid search for simplicity, as the main scope of this article is to introduce a framework for optimally estimating the parameters for KPLS models.

4. Simulation example

This section presents a comparison of Algorithms 2 and 3 with existing work to estimate n and σ on the basis of a simulation example. Existing methods for estimating n, predominantly proposed for KPCA, include:

- the cumulative percentage variance (CPV) method [21, 22];
- kernel parallel analysis (KPA) [23];
- the residual error (RE) technique [25]; and
- the 5-fold cross-validatory (5-fold CV) approach [26].

Existing methods for estimating σ include:

- kernel target alignment (KTA) [36];
- the feature space matrix (FSM) evaluation technique [37];
- the largest variance criteria (LVC) [38];
- the dimension and variance (DaV) technique [20];
- the sample distance (SD) method [29];
- the maximum distance to mean (MDM) approach [27];
- the sum of variable spread (SVS) method [28];
- the approach in Ref [30], *i.e.* 100 times N_x or the product of variable criterion (PVC);
- the Monte Carlo cross-validation (MCCV) technique [34];
- the median of the inverse of the sample distance (MID) criteria [39];
- a genetic algorithm (GA) approach [41, 42];
- the distance of reference samples to the furthest and nearest neighbors (DFN) approach [40]; and
- the simulate annealing (SA) method [35].

The simulation example involves a random vector $\boldsymbol{z}$ (10×1), whose elements are linear/nonlinear functions of a random vector $\boldsymbol{s}$ (3×1). To simulate measurement uncertainty and other sources of noise, the linear/nonlinear functions are superimposed by a random vector $\boldsymbol{\varepsilon}$ (10×1). Eq (15) describes the linear and nonlinear relationships between the random vectors $\boldsymbol{z}$, $\boldsymbol{s}$ and $\boldsymbol{\varepsilon}$:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_1 + \sin(s_2) + s_3 \\ \tan(s_1) + s_2 + s_3^3 \\ s_1 + \tan(s_2) + s_3 \\ s_1 + s_2^3 + \sin(s_3) \\ s_1^3 + s_2 + \sin(s_3) \\ \tan(s_1) + s_2^3 + s_3 \\ s_1^3 + \tan(s_2) + s_3 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \end{pmatrix} \tag{15}$$

Each element of the random vector $\boldsymbol{s}^T = \begin{pmatrix} s_1 & s_2 & s_3 \end{pmatrix}$ has a uniform distribution such that $-1.26 \leq s_i \leq 1.26$ and the random vector $\boldsymbol{\varepsilon}$ has the normal distribution $\boldsymbol{\varepsilon} \sim \mathcal{N}\{\boldsymbol{0}, 0.1\boldsymbol{I}\}$. The KPCA model in this study was based on the random vector $\boldsymbol{z}$. For KPLS, the first three elements of $\boldsymbol{z}$ formed the random predictor vector $\boldsymbol{x}$, whilst the remaining seven random variables constituted the random vector $\boldsymbol{y}$, *i.e.* $\boldsymbol{x}^T = \begin{pmatrix} z_1 & z_2 & z_3 \end{pmatrix}$ and $\boldsymbol{y}^T = \begin{pmatrix} z_4 & z_5 & z_6 & z_7 & z_8 & z_9 & z_{10} \end{pmatrix}$.

For studying the performance of each method, a total of L = 200, 500, 1000, 2000, 4000, 5000, 6000, 7000, 8000, 10000 data points were simulated. For the introduced cross-validatory framework, we divided the data sets using m = 5, 10, 20, 50, 100, 200, 250, 500, 1000, 2000, 2500, 4000, 5000, 6000, 7000, 8000, 10000 segments wherever L/m did not produce a division remainder. For determining a minimum for the objective functions in Eqs (12) and (13) — for KPCA — and Eq (14) — for KPLS —, we used a grid search for $0.1 \leq \sigma \leq 15$ in steps of 0.1. To get a more accurate model performance, we reduced the distance between two grid points to 0.01 around the minima. Moreover, we selected $n_{max} = 8$ for Algorithm 1 in Ref [45] and Algorithm 3. All other methods listed above were implemented as described in the references cited. To compare the performance of the resultant KPCA and KPLS models, we used the following criteria:

$$e_n = \frac{J(n) - J_{opt}(n, \sigma)}{J_{opt}(n, \sigma)} 100\% \tag{16a}$$

$$e_\sigma = \frac{J(\sigma) - J_{opt}(n, \sigma)}{J_{opt}(n, \sigma)} 100\% \tag{16b}$$

Here, $J(n)$ and $J(\sigma)$ represent the objective functions in Eqs (12) to (14). These values are also cross-validated for $m = L$, *i.e.* leave-one-out cross-validation. Moreover, $J_{opt}(n, \sigma)$ is the optimal value for J obtained by applying Algorithm 1 in Ref [45] and Algorithm 2 for KPCA, and Algorithm 3 for KPLS. $J_{opt}(n, \sigma)$ is also evaluated for $m = L$. Subsections 4.1 and 4.2 summarize the results of this comparison for KPCA and KPLS, respectively.

4.1. Comparing accuracy of identified KPCA models

Table 1 summarizes the results of this comparison for $L = 5000$ simulated data points. For estimating n_f, Algorithm 1 in Ref [45] correctly estimated $n_f = 3$. In contrast, the CPV, KPA, RE and 5-fold CV methods substantially overestimated this number. Particularly the CPV and KPA approaches produced implausible estimates that well exceeded $N_z = 10$. By examining the working of the CPV, KPA, RE and the 5-fold CV methods, the first three ones rely on user defined thresholds, which renders them subjective. The 5-fold CV technique assesses the performance of the KPCA model on independent data and does, therefore, not require setting any threshold. However, together with the CPV, KPA and RE methods, the 5-fold CV technique requires a preestimate of σ_f and hence, none of them estimate n_f optimally. Note that σ_f was selected to be 8.25, which Figure 1(a) shows to be the optimal value computed by Algorithm 2.

For estimating σ_f, we selected $n_f = 3$ for each method in order to guarantee a fair comparison. In practice, however, neither an optimal estimate for n_f nor σ_f is available. Given that existing work may yield and overestimate, as highlighted in the preceding analysis of this simulated data set, this concludes that σ_f may not be optimal. Even when selecting $n_f = 3$, Table 1 summarizing the estimates for σ_f using the competitive KTA, FSM, LVC, MID, DFN, GA, MDM, SVS, SD and PVC techniques and indicates these methods yield suboptimal estimates. Whilst the estimate of the SD techniques was close to that obtained by applying Algorithm 2, all other estimates were significantly different.

To assess the effect of the estimates of σ_f upon the accuracy of the resultant KPCA models, we utilized Eq (16b) for $J_{opt} = J(n_f = 3, \sigma_f = 8.25)$. For all $J(\sigma_f)$-values, we selected $n_f = 3$ and the σ_f computed by each of the competitive methods. The resultant e_{σ_f} values, listed in Table 1, confirm that the estimate of SD method produced a comparable performance of the corresponding KPCA model. All other estimates produced considerably less accurate KPCA models. More precisely, the relative performance difference using the estimate of the KTA, FSM and MID techniques exceeded 600%. For all other methods, the relative difference is in excess of 15%.

Next, we examined the effect of various n_f using Eq (16a) upon the accuracy of the corresponding KPCA model. For this, we computed $J(n_f)$ by selecting $\sigma_f = 8.25$ and listed the e_{n_f} values in Table 1. It is interesting to note that the departures, even for the implausible estimates of the CPV and KPA methods, were not as pronounced as it was the case for the estimates of σ_f. This implies that it is of fundamental importance (i) to simultaneously estimate them, *i.e.* not fixing one and only determine the other one, and (ii) to acknowledge that an incorrectly estimated σ_f-value can have a profound and undesired impact upon the accuracy of the corresponding KPCA model.

By reexamining Figure 1(a), the value for $J_{opt} = J(n_f = 3, \sigma_f = 8.25) = 0.118$. For $L \to \infty$, we can conclude that $J_{opt} \to 0.1$, which follows from the fact that $\boldsymbol{\varepsilon} \sim \mathcal{N}\{\mathbf{0}, 0.1\boldsymbol{I}\}$. With an increasing number of data points, Figure 1(b) confirms that the computed value of J_{opt} decreases and converged to the theoretical threshold of 0.1. Finally, we examined how the optimal σ_f depended on the number of data points. Figure 1(c) indicates that the larger the sample size the larger σ_f became. Between the range $2000 \leq L \leq 10,000$, the empirical relationship between L and σ_f was determined to be $\sigma_f = 0.002 \times L + 7.3922$. For smaller sample sizes, $L < 2000$, the estimated σ_f reduces by a significantly larger slope.

To demonstrate the importance of independently assessing the performance of the identified KPCA models, we also applied Eqs (16a) and (16b) based on models that were identified using the entire data set. This implies that the comparison between the models obtained based on parameters suggested by existing methods and the optimal framework was not cross-validated. By setting $\sigma_f = 8.25$ and selecting $n_f = 5, 6, 17$ and 21, the four resultant models showed a better performance than that constructed for $n_f = 3$. This is not surprising, as $n_f \to N$ implies that $e_n \to -100\%$. This, however, does not reveal the correct size of $\boldsymbol{s}$, *i.e.* $n_f = 3$. By setting $n_f = 5$ and selecting the values for σ_f suggested by the various competitive techniques, the resultant models showed a better performance if $\sigma_f > 8.25$. Conversely, utilizing smaller kernel parameters produced KPCA models that are not as accurate. This is also not surprising, as $\sigma_f \to \infty$ yield $K_f(\boldsymbol{z}_i, \boldsymbol{z}_j) \to 1$, which follows from Eq (4). This, in turn, yields a good performance to

reconstruct the recorded data points but results in a poor generalization, which can be noticed by directly comparing the performance with the cross-validated one.

Finally, the application of Algorithm 2 to estimate σ_f took 87 minutes, similar to the running time of the GA technique. For this comparison, each method was implemented in Matlab$^{\text{TM}}$, version 7.11.0, and the computation was performed on a 64-bit Windows 10 Pro operating system, 8GB of memory(RAM) and an Intel® Core$^{\text{TM}}$ i7-4700HQ clocked at 2.40GHz. We like to note that integrating Algorithm 2 into a commercial software package would require a fraction of the reported running time. This is because Matlab$^{\text{TM}}$ is not a compiled language and, hence, significantly slower than directly compiled code.

4.2. Comparing accuracy of identified KPLS models

The application of Algorithm 3 produced an estimation of $n_h = 3$ and $\sigma_h = 6.03$. To guarantee a fair comparison, we selected $\sigma_h = 6.03$ to estimate n_h using the 5-fold CV technique, which yielded a minimum of $n_h = 4$. With $N_x = 3$, however, the estimate of $n_h = 4$ is implausible. Nonetheless, we utilized Eq (16a) for $m = L = 5000$, and defined $J_{opt} = J(n_h = 3, \sigma_h = 6.03) = 0.1014$ and $J = J(n_h = 4, \sigma_h = 6.03)$, to determine relative difference in performance, which was 3.06%. This confirms (i) that Algorithm 3 computed a minimum of the objective function $J(n_h, \sigma_h)$ and (ii) that limiting $m = 5$ may not yield and optimum in a cross-validatory sense although the difference was not very significant. More importantly, however, is the fact that we preselected $\sigma_h = 6.03$. Figure 1(b) and Table 1 highlight that a non-optimal selection of σ_h can yield less accurate predictions by the identified KPLS models. This is examined next.

Table 1 lists the estimates of σ_h for the KTA, FSM, LVC, MID, GA, DaV, MCCV and SA methods. To guarantee a fair comparison, we selected $n_h = 3$. The competitive methods produced a wide range of estimates, ranging from 0.1 (KTA) to 8.9 (LVC). To assess the impact of the individual estimates of σ_h upon the accuracy of the corresponding KPLS models, we applied Eq (16b) for $J_{opt} = J(n_h = 3, \sigma_h = 6.03)$ and listed the e_{σ_h} values in Table 1. Whilst the estimates of σ_h using the LVC, MID, GA, DaV ($r = 10$) and the MCCV techniques did not result in KPLS models that have a significantly less accurate prediction, the estimates by the remaining methods produced substantially less accurate KPLS models. Note that the effect of varying σ_h between 5.1 and 8.9, these being the estimates of the MCCV and the LVC methods, respectively, did not have a profound impact of the resultant models, which can also be noted by examining Figure 1(b).

To empirically verify the need for an independent performance assessment, we also applied Eqs (16a) and (16b) to the residuals obtained directly from the identified models, *i.e.* the residuals were not determined in a cross-validatory fashion. Similar to the KPCA model, increasing the number of latent components by keeping the kernel parameter unchanged can only reduce the model error. On the other hand, fixing $n_h = 3$ and comparing the performance of the resulting KPLS models with the one based on the optimal parameter set suggests that there is hardly any difference unless the kernel parameter is considerably smaller.

Recall that we preselected $n_h = 3$. Practically, an optimal estimate of n_h may not be available, which underpins the necessity of simultaneously estimating n_h and σ_h to guarantee the identification of an optimal KPLS model. This particular issue is studied in more detail in the next section, which applies each of the methods to recorded/experimental data sets. Finally, the application of Algorithm 3 took 53 minutes, which is comparable with the time consumed by the GA and the MCCV methods. Given that Matlab$^{\text{TM}}$ is a compiled language, embedding Algorithm 3 into a commercial software package would require a fraction of the running time reported here.

[Table 1 about here.]

[Figure 1 about here.]

5. Application studies

This section presents the application of the cross-validatory framework as well as the CPV, KPA, RE, 5-fold CV, KTA, FSM, LVC, DaV, SD, MDM, SVS, PVC, MCCV, MID, GA, DFN and SA methods to estimate n and σ for a total of three recorded data sets. These sets stem from two processes in the chemical industry, *i.e.* a glass melter and a distillation process, and a set of lab data from a mixing experiment. Different from the previous section, we do not know the optimal number of latent component sets. Subsections 5.1

and 5.2 summarize the results of this comparison based on the recorded data from the glass melter and the distillation processes, respectively. Subsection 5.3 presents the results of comparing each method on the basis of the experimental data from the mixing experiment. Finally, Subsection 5.4 summarizes the results of each application study. Before identifying KPCA and KPLS models, each data set was normalized, *i.e.* each variable was mean centered and scaled to unit variance. Moreover, each model was based on Gaussian kernels, requiring the estimation of the kernel parameter and the number of latent component sets. For the application of the MCCV method, we randomly left 5 samples out for each cross validation procedure and the number of Monte Carlo runs was 1000.

5.1. Accuracy comparison — glass melter process

A process description is given first, followed by comparing the performance of the identified KPCA and KPLS models for each method.

5.1.1. Process description

This process is part of a disposal procedure for waste material. The waste is constantly introduced to the melter vessel in the form of a powder. The powder is clad in molten glass, which is discretely introduced. The powder and raw glass mixture is heated by four induction coils that are positioned around the vessel. The constant filling results in an increasing liquid level. When the liquid column reaches a certain height, the melter is emptied and a new cycle of filling and heating begins. From this process, a total of 21 variables were recorded at a sampling interval of 5 minutes, including 15 temperatures inside the vessel, the power in the 4 induction coils, the voltage applied to the induction coils and the viscosity of the molten glass. A total of 7500 data points of the random vector $\boldsymbol{z}$ (21×1) were used to estimate n_f and σ_f for KPCA models using the different methods studied here. For KPLS, the random vector $\boldsymbol{x}$ (5×1) contained the measurements of the four induction coils and the voltage applied to the coils, whilst the random vector $\boldsymbol{y}$ (16×1) contained the 15 temperature readings and the viscosity of the molten glass.

5.1.2. Comparing accuracy of identified KPCA models

Before identifying KPCA models, the data set was divided into $m = L = 7500$ segments. To optimally estimate n_f and σ_f, we used a grid search and defined the step size to be initially 0.1. We then refined the search using the smaller step size of 0.01 around the smallest values of $J_f(n_f, \sigma_f)$. The cross-validatory framework for KPCA estimated $n_f = 5$ and $\sigma_f = 32.14$. This resulted in $J_{opt} = J_f(n_f = 5, \sigma_f = 32.14) = 0.2228$.

Table 2 shows the estimates of n_f obtained by the CVP, KPA, RE and 5-fold CV techniques. We selected $\sigma_f = 32.14$ to guarantee a fair comparison. The estimates ranged from $n_f = 6$ (5-fold CV) to $n_f = 18$ (KPA). Each of these estimates produce a less accurate KPCA model when compare to that obtained by the cross-validatory framework. More precisely, utilizing Eq (16a) resulted in relative performance deteriorations that can exceed 15%.

The estimates of σ_f when using the KTA, FSM, LVC, MID, DFN, GA, MDM, SVS, SD and PVC methods are also summarized in Table 2, ranging from 0.08 (MID) to 2100 (PVC). Recall that we preselected $n_f = 5$. On the basis of Eq (16b), it can be concluded that larger values of the kernel parameter, *e.g.* suggested by the GA, MDM and PVC techniques did not yield a considerably less accurate KPCA model, as the relative difference was only a few percent. Conversely, smaller values, *e.g.* those computed by the KTA, FSM, LVC, MID and DFN methods, produced vastly inferior KPCA models. The application of the SD method also produced a much smaller estimate of σ_f but the relative difference in model performance to the optimal one is just over 3%. Note that the methods suggesting a more suitable σ_f rely on quantifying the "dispersity" of the data points in the original variable space, *i.e.* the MDM or the SVS techniques, or utilize the accuracy of the KPCA model (GA method). Figure 2(a) confirms the observations that any σ_f value that is between 28 to 36 resulted in a comparable performance of the corresponding KPCA model.

5.1.3. Comparing accuracy of identified KPLS models

The optimal estimates of n_h and σ_h for applying Algorithm 3 were 5 and 0.62, respectively, which yielded $J_{opt} = J_h(n_h = 5, \sigma_h = 0.62) = 0.2078$. Selecting $\sigma_h = 0.62$, the application of the 5-fold CV approach suggested $n_h = 7$ which is implausible, given that $\boldsymbol{x} \in \mathbb{R}^5$. Eq (16a) highlights that $n_h = 7$ reduces the predictive performance of the identified KPLS model by around 18%. Table 2 lists the estimates of σ_h for

which we preselected $n_h = 5$. Seven of the competitive methods produced significantly larger estimate for the kernel parameter. Sufficiently close estimates to the optimal value of $\sigma_h = 0.62$ were suggested by the MID, the MCCV and the GA method. The application of Eq (16b) showed that selecting a non-optimal kernel parameter can result in a relative reduction in predictive performance of up to 50%. Conversely, the application of the MID, MCCV and SA methods allowed constructing KPLS model that had a comparable performance. However, the optimal number of latent variable sets, *i.e.* $n_h = 5$, was predetermined, which is generally unavailable. This issue is explored in more detail in the next subsection, which analysis a process that contains a larger predictor variable set.

[Table 2 about here.]

5.2. Accuracy comparison — distillation process

A description of the process is given first. This is followed by summarizing the results of comparing the performance of the identified KPCA and KPLS models in two separate subsections.

5.2.1. Process description

The process purifies butane from a fresh feed that is composed of propane (C3), butane (C4) and pentane (C5). A detailed description for this process can be found in Chapter 5 in Ref [6]. From this process, a total of 12 variables were recorded, 3 temperatures at different trays (Tray 2, 14, 31), the fresh feed and reboiler temperature, flow rates of fresh feed, top and bottom product flow, the reboiler steam flow, and 3 concentrations (percentages C3 in C4, C5 in C4 and C4 in C5). Data points were recorded at a sampling interval of 30 seconds. To identify KPCA and KPLS models, a total of 7500 data points of the random vector $\boldsymbol{z}$ (12×1) were recorded, covering a continuous period of around 62 hours. Each KPCA model extracted the latent variables of the random vector $\boldsymbol{z}$. For identifying KPLS models, the random vector $\boldsymbol{z}$ was then divided into the $\boldsymbol{x}$ (7×1), which included the three tray temperature variables, fresh feed flow and temperature and the reboiler steam flow and temperature, and $\boldsymbol{y}$ (5×1) containing the remaining variables.

5.2.2. Comparing accuracy of identified KPCA models

The application of Algorithm 1 in Ref [45] and Algorithm 2 estimated n_f and σ_f to be 5 and 16.08, respectively. This resulted in a $J_{opt} = J(n_f = 5, \sigma_f = 16.08) = 0.0531$. By preselecting the optimal value of 16.08 as σ_f, the application of the CPV, KPA, RE and 5-fold CV methods produced the estimates listed in Table 3, which were substantially larger. By applying Eq (16a), the relative prediction accuracy of the resultant KPCA models was reduced by up to 10%, confirming that existing work cannot optimally determine the number of principal components.

Next, Table 3 shows that applying the KTA, FSM, LVC, MID, DFN, GA, MDM, SVS, SD and PVC methods produced estimates ranging from 0.1 to 1200. As before, we preselected $n_f = 5$ to guarantee that each corresponding KPCA model is based on the same number of principal components. Using Eq (16b) to assess the impact of σ_f upon the relative prediction accuracy confirms that, as before, the larger estimates of σ_f by the GA, the MDM, the SVS and the PVC techniques did not yield a substantially compromised performance accuracy. In sharp contrast, estimates that are substantially smaller than 16.08, *i.e.* those suggested by the KTA, FSM, LVC, MID and DFN methods resulted in substantially less accurate KPCA models.

5.2.3. Comparing accuracy of identified KPLS models

The application of Algorithm 3 estimated n_h and σ_h to be 5 and 3.1, respectively, producing $J_{opt} = J(n_h = 5, \sigma_h = 3.1) = 0.2203$. Given that the objective function, $J(\cdot)$, is based on the average prediction accuracy of KPCA/KPLS models, it is interesting to note that the prediction of $\boldsymbol{y}$ (5×1) using $\boldsymbol{x}$ (7×1) is less accurate than the contribution of the extracted n_f principal components in reconstructing $\boldsymbol{z}$ (12×1). This indicates that the top and bottom flow and to a lesser extend the three concentrations could still be accurately predicted. However, by comparing the residual variance of the concentration measurements for the optimal KPCA and KPLS models, they are between 0.1 to 0.3 and hence, substantially larger than those of the remaining variables. Table 3 shows that the 5-fold CV method, based on $\sigma_h = 3.1$, yielded an estimate of $n_h = 8$, which is implausible given that $\boldsymbol{x} \in \mathbb{R}^7$ although the relative loss of predictive power is only 0.64%.

By preselecting $n_h = 5$, Table 3 highlights that the KTA, FSM, LVC, MID, DFN, GA, DaV, MCCV and SA methods suggested values within the range $0.5 \leq \sigma_h < 9$. Methods that suggested σ_h values that yielded KPLS models showing a comparable performance to the optimal one include the MID, MCCV and DaV (for $r = 1$ and 2). More precisely, the suggested σ_h values are close the optimal one of 3.1. Different from the KPCA models, where substantially larger values or values that are close to $\sigma_f = 32.14$ still produced a performance that is comparable to the KPCA model based on the optimal set of parameters, any deviation from $\sigma_h = 3.1$ produced KPLS models that showed a considerably less accurate performance, which can well exceed 50%.

[Table 3 about here.]

5.3. Accuracy comparison — mixing experiment

This application study involves experimental data from a mixing experiment, which is detailed first. Different to the previous studies, for which the number of recorded data points was substantially larger than the size of the random vector $\boldsymbol{z}$, the number of recorded data points in this application study was significantly smaller than the number of recorded variables. As before, two separate subsections summarize the results of comparing the individual KPCA and KPLS models.

5.3.1. Description of the mixing experiment

In this experiment, 100 ml of H_2O was added to 100 ml of Isopropyl Alcohol (IPA) over a period of around 17 minutes. For the first 12 minutes, H_2O was initially added at a rate of 100 ml/h. After that, the flowrate of water increased to 1 l/h over a period of around 4 minutes and 30 seconds to amount to the total of 100 ml of added water. A data point of this solution was taken every 30 seconds to determine the concentrations of IPA and H_2O as well as its Raman spectroscopy spectra, containing 1476 intensities. The Raman spectra were computed by an Avalon Instruments RamanStation R3. The device uses a 785 nm wavelength, 110 mW laser and CCD detector, coupled with a fiber optic probe encased in a high pressure sheath with a sapphire window.

5.3.2. Comparing accuracy of identified KPCA models

The application of Algorithm 1 in Ref [45] and Algorithm 2 estimated n_f and σ_f to be 5 and 715.2, respectively, resulting in a $J_{opt} = J(n_f = 5, \sigma_f = 715.2) = 0.1355$. In sharp contrast, Table 4 confirms that the CPV, KPA, RE and 5-fold CV methods suggested substantially larger estimates. As before, σ_f was selected as the optimal estimate of 715.2. Given that the changes in the intensity profiles and the concentrations of H_2O and IPA were mainly driven by adding H_2O, the number of latent components is expected to be small. Despite the larger estimates of n_f, however, the relative difference in accuracy were in the range of 6 to 12%.

The estimation of σ_f for $n_f = 5$ using the KTA, FSM, LVC, MID, DFN, GA, MDM, SVS, SD and PVC methods yielded values of $8 \times 10^{-4} \leq \sigma_f \leq 1.5 \times 10^5$. In a similar fashion to the previous application studies, values that are close to the optimal estimate or substantially larger values of σ_f produced KPCA models that had a comparable accuracy to that based on the optimal values for n_f and σ_f. Conversely, significantly smaller values led to KPCA models that had a considerably less accurate performance, as evaluated by Eq (16b). As for the previous two applications, it can be observed that the MDM, SVS and PVC methods produced estimates that yielded accurate KPCA models. In addition to that, the SD method produced an estimate that is relatively close to the optimal one. All other methods did not provide sensible estimates. Again, note that each method was furnished with an optimal estimate of n_f, which would normally not be available.

The analysis of this data set in Ref [45] outlined that there are only 608 out of the 1476 intensities which showed to have common trends. Based on this, we reduced the entire set of 1476 intensities plus the concentrations of H_2O and IPA to a total of 610 variables and reapplied Algorithm 1 in Ref [45] and Algorithm 2. This produced the optimal estimates of $n_f = 4$ and $\sigma_f = 301.52$, and reduced the optimal value of the objective function from 0.1355 to 0.0128.

5.3.3. Comparing accuracy of identified KPLS models

Applying Algorithm 3 to estimate the number of latent variable sets and the kernel parameter suggested $n_h = 5$ and $\sigma_h = 30.9$, which yielded $J_{opt} = J(n_h = 5, \sigma_h = 30.9) = 0.007$. By pre-selecting $\sigma_h = 30.9$, the 5-fold CV method estimated n_h to be 16, which is considerably higher than the optimal estimate of $n_h = 5$. This overestimate manifested itself in a compromised relative performance difference of almost 20%, which is undesirable.

Pre-selecting $n_h = 5$, Table 4 highlights that using the KTA, FSM, LVC, MID, DFN, GA, DaV, MCCV and SA methods to estimate σ_h produced values in the range $8 \times 10^{-4} \leq \sigma_h \leq 125$. In fact, only the LVC and the MCCV technique led to KPLS models that showed a comparable performance to the optimal KPLS one. With the exception of the KPLS model that was based on the estimate of the DFN method, the remaining KPLS models had a very poor relative performance.

Based on the reduced set of 608 intensities to predict the concentration of H_2O and IPA, the application of Algorithm 3 estimated n_h and σ_h as 4 and 20.21, respectively. This reduced the optimal value of the objective function from 0.007 to 1.69×10^{-3}.

[Table 4 about here.]

5.4. Summary of application studies

Comparing the results for each application study, it can be noticed that the algorithms of the cross-validatory framework yielded the most accurate KPCA and KPLS models. The merits of this framework rely on the assessment of model accuracy based on data points that were not included in the model identification stages. More precisely, the evaluation of the model performance was independent of the identification of the KPCA and KPLS models. Moreover, utilizing the objective functions in Eqs (12) to (14) is in line with the properties of PCA/PLS and KPCA/KPLS. For PCA, each principal component extracts the maximum amount of variance from the random vector $\boldsymbol{z}$ and minimizes the residual variance of $\boldsymbol{\varepsilon} = \boldsymbol{z} - \widehat{\boldsymbol{z}}$ [6]. The same holds true for KPCA being a generic extension for PCA [16]. The PLS/KPLS techniques produce regression models with the aim of predicting a response set $\boldsymbol{y}$ as accurately as possible based on a predictor set $\boldsymbol{x}$. This necessitates estimating n and σ on the basis of model accuracy.

For the number of latent components, i.e. n_f for KPCA and n_h for KPLS, each of the existing techniques proposed in the literature suggest a non-parsimonious estimate, *i.e.* they overestimated this number. This has to be seen in relation to the fact that each method was given the optimal estimate of the kernel parameter, *i.e.* σ_f for KPCA and σ_h for KPLS, to guarantee a fair comparison.

For the kernel parameter, existing methods produced vastly different estimates for each application. It was interesting to note that the overestimates of n_f and n_h did not yield KPCA and KPLS models that had a substantially poorer performance, although the relative deviation in performance accuracy could exceed 10%. Conversely, an incorrect estimate of the kernel parameter had a very profound impact upon the relative performance accuracy. Methods which produced estimates of σ_f that resulted in KPCA models that had a comparable performance were the GA, MDM, SVS and PVC ones. With the exception of PVC, which is essentially a simple *ad hoc* rule, MDM and SVS are methods that examine the dispersity of the data points within the data space and the GA technique relies on an optimal estimate of the model performance. However, each of these methods requires a pre-estimate of n_f, which is usually not available, and do not rely on an independent assessment of the model performance.

For KPLS, only the MCCV technique produced comparable estimates of σ_h compared to the optimal cross-validatory framework for each application study. None of the other methods showed, consistently, a comparable performance and yielded estimates that resulted in considerably less accurate KPLS models, which is undesireable. The MCCV method, however, also requires a pre-estimate of n_h, which is practically not available. In addition to that, the MCCV method relies, principally, on the same cross-validatory approach as the introduced framework. However, the Monte-Carlo approach is computationally more expensive than the proposed framework and it is possible that some data points are not utilized at all or utilized several times for model identification and the performance assessment of the identified model.

[Figure 2 about here.]

Figures 2(a) to 2(c) show the dependency of σ_f and σ_h upon $J(\cdot)$ for the glass melter and the distillation processes as well as the mixing experiment, respectively. Each plot confirms the preceding observations that

the resulting functions are convex in the vicinity of the optimal kernel parameters, with the exception of the minor kink around $\sigma_h = 3.3$ for J_h in Figure 2(b). In addition to that, Figures 3(a) and 3(b) show how $J(\cdot)$ varies with n. The dependency of the average prediction accuracy for KPCA and KPLS models upon n_f and n_h, respectively, is a convex function. Both, Figures 2 and 3 confirm that $J(n, \sigma)$ are convex functions in the vicinity of its minimum. This, in turn, confirms that the cross-validatory framework, introduced in this article, is optimal for simultaneously estimating n and σ. In sharp contrast, existing work cannot be seen as optimal and relies on *ad hoc* rules and/or only estimating one parameter.

[Figure 3 about here.]

Next, using the model residuals, *i.e.* omitting a cross-validatory assessment, based on the identified models for Eqs (16a) and (16b) yielded the following. The KPLS models showed similar performance for a fixed n_h and a varying σ_h. A notable exception is for the KPLS model of the distillation unit that was based on the relatively small $\sigma_h = 0.2354$. Different to the KPLS models, the KPCA models showed more substantial variations in performance. Increasing the number of latent components produced better KPCA models, which is expected. Conversely, varying σ_h by keeping n_h constant did only produce marginally improved models. More precisely, considerably smaller values compared to the optimal kernel parameter yielded significantly less accurate models. For the recorded data of the mixing process, the KPLS models showed a more pronounced effect to variations in the model parameters. Recall that it is advisable to rely on an independent assessment of the model, *i.e.* utilizing the cross-validatory framework.

Finally, Table 5 lists the optimal estimates for each of the three application studies including the running time. Subsection 4.1 provides details on how the computations were carried out. The running time of the methods that are embedded within the cross-validatory framework is similar to that of the MCCV and GA methods but significantly longer than the remaining techniques. However, existing work produced substantial overestimates for n and suggested kernel parameters that produced KPCA and KPLS models which had substantially less accurate prediction accuracies. In addition to that, given that Matlab$^{\text{TM}}$ is an interpreted language incorporating the proposed cross-validatory framework into a commercial software package reduces the running time to a fraction of what is reported in Table 5.

[Table 5 about here.]

6. Concluding summary

This article has introduced a framework for optimally estimating the number of latent components, n, and the kernel parameter, σ, for constructing kernel principal component analysis and kernel partial least squares models. Both methods are of fundamental importance in chemometric applications as they represent generic nonlinear extensions to their linear counterparts. The revision of existing work in estimating these parameters showed that they assume that one parameter is known or predefined and the second parameter is estimated mostly by applying *ad hoc* rules with few utilizing objective functions to compute optimal estimates. To find an optimal estimate of both parameters in an effective and statistically independent fashion has, consequently, not been proposed and has been the motivation for the presented work.

The framework introduced here utilizes various cross-validatory schemes to guarantee that both parameters have been estimated optimally with respect to various objective functions, detailed in Section 3. These are based on the modeling accuracy. The framework has first been benchmarked against existing estimation methods on the basis of a simulation example. To obtain KPCA and KPLS models this examples relied on $N_z = 10$, and $N_x = 3$ and $N_y = 7$ variable sets, respectively, that are driven by 3 latent sets. The cross-validatory framework produced a correct estimation of $n = 3$ latent variable sets for kernel principal component analysis and kernel partial least squares. In sharp contrast, existing methods produced overestimates of this number. For estimating the kernel parameter, the cross-validatory framework yielded optimal estimates for both kernel methods, whilst only few methods produced similar estimates when the correct number of latent components has been predetermined. Conversely, all other method yielded non-optimal estimates.

The article has also contrasted the introduced cross-validatory framework with existing work on the basis of three different data sets. These are recorded data from two chemical processes (many samples and few variables) and a data set from a mixing experiment (few samples and many variables). The application of

each method to the data from the two chemical processes has shown that the cross-validatory framework estimated the smallest number of latent components and the smallest prediction error for both kernel models, whilst most methods overestimated this number. For the data of the mixing experiment, only the cross-validatory framework produced a small and plausible estimate of the number of latent components, whilst all other method have yielded overestimates. These results directly follow from the objective functions of the cross-validatory algorithms, which estimate n and σ, such that they directly assess the impact of retaining $1 \leq \widetilde{n} \leq n$ and varying σ simultaneously.

7. Acknowledgement

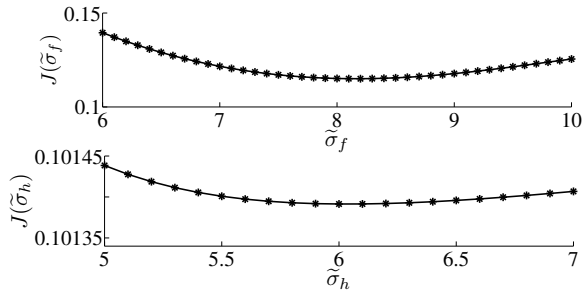
Yujia Fu is grateful for the financial support from the Fundamental Research Funds for Central Universities (No. JUSRP51510), National Natural Science Foundation of China (No. 61273070), 111 project (No. 12018) and Jiangsu College Graduate Research Innovation Project (No. KYLX15-1168).

References

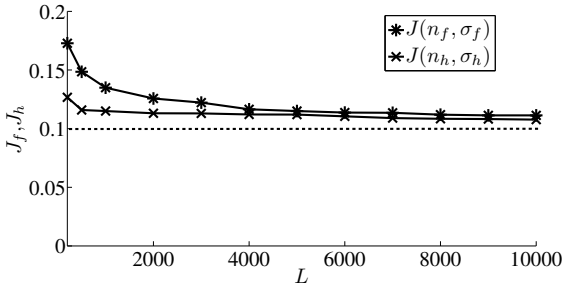
[1] A. Lombardo, O. Schifanella, A. Roncaglioni, E. Benfenati, Quantitative Structure-Activity Relationship (QSAR) in Ecotoxicology, Springer Netherlands, 2013.
[2] M. Chromčikova, V. Zemanová, A. Plško, B. Hruška, T. Gavenda, Thermodynamic model and raman spectra of CaO-P_2O_5 glasses, Journal of Thermal Analysis and Calorimetry 121 (1) (2015) 269–274.
[3] Z. Wu, H. Li, D. Tu, Application of fourier transform infrared (FT-IR) spectroscopy combined with chemometrics for analysis of rapeseed oil adulterated with refining and purificating waste cooking oil, Food Analytical Methods 8 (10) (2015) 2581–2587.
[4] J. F. Ruiz, A. Maña, C. Rudolph, An integrated security and systems engineering process and modelling framework, Computer Journal 58 (10) (2015) 2328–2350.
[5] O. Taylan, D. Kaya, A. Demirbas, An integrated multi attribute decision model for energy efficiency processes in petrochemical industry applying fuzzy set theory, Energy Conversion & Management 117 (2016) 501–512.
[6] U. Kruger, L. Xie, Advances in statistical monitoring of complex multivariate processes: with applications in industrial process control, John Wiley & Sons, Chichester, U.K., 2012.
[7] J. Zeng, U. Kruger, J. Geluk, X. Wang, L. Xie, Detecting abnormal situations using the Kullback-Leibler divergence, Automatica 50 (11) (2014) 2777–2786.
[8] L. Xie, J. Zeng, U. Kruger, X. Wang, J. Geluk, Fault detection in dynamic systems using the Kullback-Leibler divergence, Control Engineering Practice 43 (2015) 39–48.
[9] E. R. Malinowski, Factor Analysis in Chemistry, 3rd Edition, John Wiley & Sons, New York, 2002.
[10] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and Intelligent Laboratory Systems 2 (1-3) (1987) 37–52.
[11] P. Geladi, Notes on the history and nature of partial least squares (PLS) modelling, Journal of Chemometrics 2 (4) (1988) 231–246.
[12] M. A. Kramer, B. L. Palowitch, A rule-based approach to fault diagnosis using the signed directed graph, AIChE Journal 33 (7) (1987) 1067–1078.
[13] B. Kegl, A. Krzyzak, T. Linder, Z. K., Learning and design of pricipal curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (3) (2000) 181–297.
[14] S. J. Qin, T. J. McAvoy, Nonlinear PLS modeling using neural networks, Computers & Chemical Engineering 16 (4) (1992) 379–391.
[15] E. C. Malthouse, A. C. Tamhane, R. S. H. Mah, Nonlinear partial least squares, Computers & Chemical Engineering 21 (8) (2010) 875–890.
[16] U. Kruger, J. Zhang, L. Xie, Developments and applications of nonlinear principal component analysis – a review, in: Principal Manifolds for Data Visualization and Dimension Reduction, Springer, 2008, pp. 1–43.
[17] B. Schölkopf, A. Smola, K. R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.
[18] B. Schölkopf, A. Smola, K. R. Müller, Kernel principal component analysis, in: Artificial Neural Networks-ICANN'97, Springer, 1997, pp. 583–588.
[19] R. Rosipal, L. J. Trejo, Kernel partial least squares regression in reproducing kernel Hilbert space, The Journal of Machine Learning Research 2 (2002) 97–123.
[20] K. Kim, J. M. Lee, I. B. Lee, A novel multivariate regression approach based on kernel partial least squares with orthogonal signal correction, Chemometrics and Intelligent Laboratory Systems 79 (1) (2005) 22–30.
[21] J. H. Cho, J. M. Lee, S. W. Choi, D. Lee, I. B. Lee, Fault identification for process monitoring using kernel principal component analysis, Chemical Engineering Science 60 (1) (2005) 279–288.
[22] R. Zhang, W. Wang, Y. Ma, Approximations of the standard principal components analysis and kernel PCA, Expert Systems with Applications 37 (9) (2010) 6531–6537.
[23] K. W. Jorgensen, L. K. Hansen, Model selection for gaussian kernel PCA denoising, IEEE Transactions on Neural Networks and Learning Systems 23 (1) (2012) 163–168.
[24] W. Ku, R. H. Storer, C. Georgakis, Disturbance rejection and isolation by dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 30 (1995) 179–196.

[25] Y. Xiao, H. Wang, W. Xu, Model selection of gaussian kernel PCA for novelty detection, Chemometrics and Intelligent Laboratory Systems 136 (2014) 164–172.
[26] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, Q. M. Gu, A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine, Neurocomputing 55 (1) (2003) 321–336.
[27] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, E. W. Lang, KPCA denoising and the pre-image problem revisited, Digital Signal Processing 18 (4) (2008) 568–580.
[28] J. Ni, C. Zhang, S. X. Yang, An adaptive approach based on KPCA and SVM for real-time fault diagnosis of HVCBs, IEEE Transactions on Power Delivery 26 (3) (2011) 1960–1971.
[29] T. Kenig, Z. Kam, A. Feuer, Blind image deconvolution using machine learning for three-dimensional microscopy, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (12) (2010) 2191–2204.
[30] X. Deng, X. Tian, Nonlinear process fault pattern recognition using statistics kernel PCA similarity factor, Neurocomputing 121 (2013) 298–308.
[31] L. J. Trejo, R. Kochavi, K. Kubitz, L. D. Montgomery, R. Rosipal, B. Matthews, Measures and models for predicting cognitive fatigue, in: Defense and Security, 2005.
[32] L. J. Trejo, K. Kubitz, R. Rosipal, R. L. Kochavi, L. D. Montgomery, EEG-based estimation and classification of mental fatigue, Psychology 6 (6) (2015) 572–589.
[33] F. Chu, F. Wang, X. Wang, S. Zhang, Performance modeling of centrifugal compressor using kernel partial least squares, Applied Thermal Engineering 44 (44) (2012) 90–99.
[34] H. Shinzawa, J. Jiang, P. Ritthiruangdej, Y. Ozaki, Investigations of bagged kernel partial least squares (KPLS) and boosting KPLS with applications to near-infrared (NIR) spectra, Journal of Chemometrics 20 (8-10) (2006) 436–444.
[35] J. M. Fonville, M. Bylesjö, M. Coen, J. K. Nicholson, E. Holmes, J. C. Lindon, M. Rantalainen, Non-linear modeling of 1 H NMR metabonomic data using kernel-based orthogonal projections to latent structures optimized by simulated annealing, Analytica Chimica Acta 705 (1) (2011) 72–80.
[36] N. Cristianini, J. Kandola, A. Elisseeff, J. Shawe-Taylor, On kernel target alignment, in: Innovations in Machine Learning, Springer, 2006, pp. 205–256.
[37] C. H. Nguyen, T. B. Ho, An efficient kernel matrix evaluation measure, Pattern Recognition 41 (11) (2008) 3366–3372.
[38] B. Yang, Y. Bu, A novel gaussian kernel paramter choosing method, in: Third International Symposium on Intelligent Information Technology Application-IITA 2009, Vol. 3, IEEE, 2009, pp. 83–86.
[39] L. Zhang, W. Zhou, P. Chang, J. Liu, Z. Yan, T. Wang, F. Li, Kernel sparse representation-based classifier, IEEE Transactions on Signal Processing 60 (4) (2012) 1684–1695.
[40] Y. Xiao, H. Wang, L. Zhang, W. Xu, Two methods of selecting gaussian kernel parameters for one-class SVM and their application to fault detection, Knowledge-Based Systems 59 (2014) 75–84.
[41] F. Chen, C. Han, Time series forecasting based on wavelet KPCA and support vector machine, in: 2007 IEEE International Conference on Automation and Logistics, IEEE, 2007, pp. 1487–1491.
[42] J. Fan, S. J. Qin, Y. Wang, Online monitoring of nonlinear multivariate industrial processes using filtering KICA-PCA, Control Engineering Practice 22 (2014) 205–216.
[43] T. M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Transactions on Electronic Computers (3) (1965) 326–334.
[44] P. Geladi, B. R. Kowalski, Partial least-squares regression: a tutorial, Analytica Chimica Acta 185 (1986) 1–17.
[45] Y. Fu, U. Kruger, Z. Li, L. Xie, J. Hahn, H. Yang, Revealing underlying nonlinear data structures using optimized kernel principal component models, Journal of Chemometrics (submitted).
[46] J. Camacho, A. Ferrer, Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects., Journal of Chemometrics 26 (7) 361–373.
[47] J. Camacho, A. Ferrer, Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: practical aspects, Chemometrics & Intelligent Laboratory Systems 131 37–50.

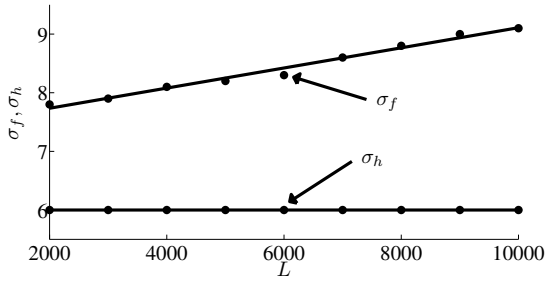
List of Figures



(a) Optimal σ for KPCA and KPLS models, cross-validatory framework for $m = L = 5000$

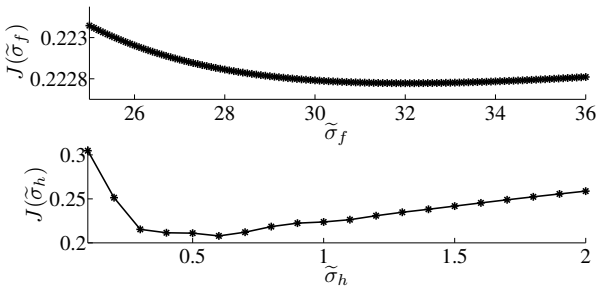


(b) Influence of $L = m$ upon $J(\cdot)$

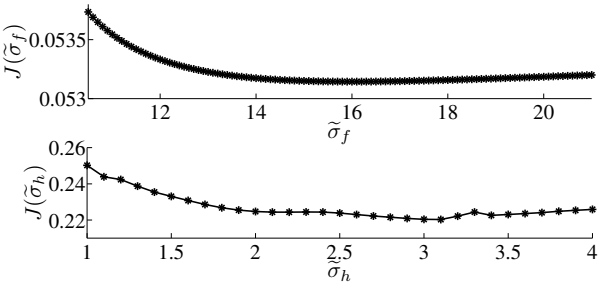


(c) Empirical dependency of σ upon L, $m = L$

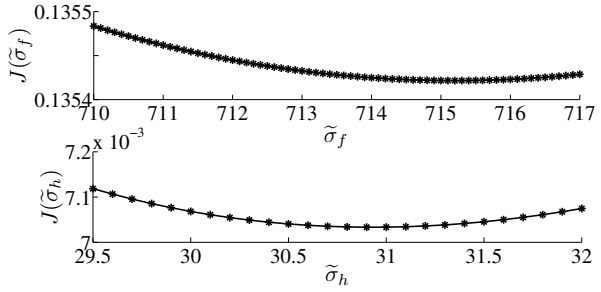
Figure 1: Estimation of σ and impact of m and L upon minimum of objective function.



(a) Subsection 5.1 – glass melter process

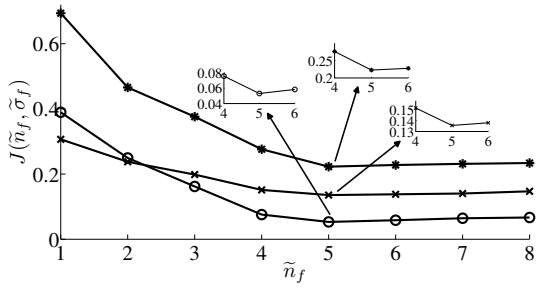


(b) Subsection 5.2 – distillation process

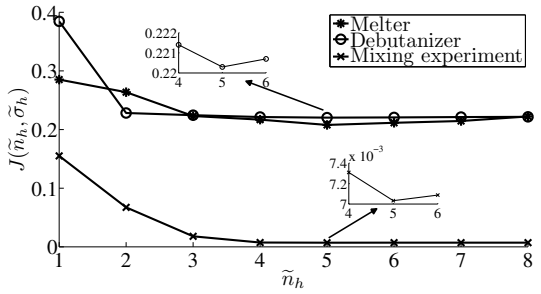


(c) Subsection 5.3 – mixing experiment

Figure 2: Optimal estimation of σ for each application study.



(a) Estimation of n_f for each KPCA model



(b) Estimation of n_h for each KPLS model

Figure 3: Optimal estimation of the number of latent variable sets.

List of Tables

Table 1: Estimation of n and σ for KPCA and KPLS models — simulation example; (– - the method is not applicable)

Method	KPCA						KPLS					
	n_f	e_{n_f}		σ_f	e_{σ_f}		n_h	e_{n_h}		σ_h	e_{σ_h}	
		CV	no CV		CV	no CV		CV	no CV		CV	no CV
Estimating n												
CPV(90%)	17	12.8%	-98.7%	–	–	–	–	–	–	–	–	–
KPA	21	16.9%	-99.4%	–	–	–	–	–	–	–	–	–
RE	6	6.17%	-40.9%	–	–	–	–	–	–	–	–	–
5-fold CV	5	5.48%	-16.6%	–	–	–	4	3.06%	-6.15%	–	–	–
Estimating σ												
KTA	–	–	–	0.3	601%	298%	–	–	–	0.1	485%	-26.1%
FSM	–	–	–	0.1	607%	311%	–	–	–	0.5	75.5%	11.4%
LVC	–	–	–	J_1:3.5	110%	80%	–	–	–	J_1:7.4	0.02%	0.01%
				J_2:4.1	92.6%	65.5%				J_2:8.9	0.06%	0.03%
MID	–	–	–	0.1	607%	311%	–	–	–	0.4358	77.1%	9.46%
DFN	–	–	–	4.7	53.1%	54.3%	–	–	–	5.7	0.01%	-0.01%
GA	–	–	–	12.45	15.7%	-36.8%	–	–	–	7.67	0.05%	0.01%
MDM	–	–	–	21.45	77%	-68.6%	–	–	–	–	–	–
SVS	–	–	–	54.43	76.5%	-71.8%	–	–	–	–	–	–
SD	–	–	–	10.1	0.97%	-13.9%	–	–	–	–	–	–
PVC	–	–	–	1000	77.4%	-71.9%	–	–	–	–	–	–
DaV	–	–	–	–	–	–	–	–	–	$1.73(r=1)$	15.7%	3.63%
										$2.45(r=2)$	4.46%	0.97%
										$3.87(r=5)$	0.42%	0.06%
										$5.48(r=10)$	0.01%	-0.01%
MCCV	–	–	–	–	–	–	–	–	–	5.1	0.03%	-0.01%
SA	–	–	–	–	–	–	–	–	–	6.04	0	-0.01%
optimal CV framework	3			8.25			3			6.03		

Table 2: Estimation of n and σ for KPCA and KPLS models — glass melter process

Method	KPCA						KPLS					
	n_f	e_{n_f}		σ_f	e_{σ_f}		n_h	e_{n_h}		σ_h	e_{σ_h}	
		CV	no CV		CV	no CV		CV	no CV		CV	no CV
Estimating n												
CPV(90%)	13	13.9%	-86.7%	–	–	–	–	–	–	–	–	–
KPA	18	18.8%	-97.8%	–	–	–	–	–	–	–	–	–
RE	9	6.81%	-62.8%	–	–	–	–	–	–	–	–	–
5-fold CV	6	2.78%	-19.9%	–	–	–	7	17.7%	-1.37%	–	–	–
Estimating σ												
KTA	–	–	–	0.9	313%	354%	–	–	–	0.1	46.7%	-2.42%
FSM	–	–	–	0.9	313%	354%	–	–	–	4.4	48.9%	6.56%
LVC	–	–	–	J_1:3.3	142%	137%	–	–	–	J_1:1.4	14.6%	2.27%
				J_2:1.4	276%	312%				J_2:0.1	46.6%	-2.42%
MID	–	–	–	0.08	349%	426%	–	–	–	0.4839	1.48%	-0.58%
DFN	–	–	–	4.1	44.8%	87.1%	–	–	–	1.7	19.8%	2.75%
GA	–	–	–	24.17	0.18%	0.36%	–	–	–	5.83	48.4%	7.1%
MDM	–	–	–	176.9	0.31%	-0.17%	–	–	–	–	–	–
SVS	–	–	–	109.01	0.27%	-0.17%	–	–	–	–	–	–
SD	–	–	–	14.92	3.23%	3.16%	–	–	–	–	–	–
PVC	–	–	–	2100	0.27%	-0.17%	–	–	–	–	–	–
DaV	–	–	–	–	–	–	–	–	–	2.24($r=1$)	28.5%	3.61%
										3.16($r=2$)	38.3%	5.64%
										5($r=5$)	47.2%	6.83%
										7.07($r=10$)	49.3%	7.38%
MCCV	–	–	–	–	–	–	–	–	–	0.6	0	-0.08%
SA	–	–	–	–	–	–	–	–	–	0.6	0	-0.08%
optimal CV framework	5			32.14			5			0.62		

Table 3: Estimation of n and σ for KPCA and KPLS models — distillation process

Method	KPCA						KPLS					
	n_f	e_{n_f}		σ_f	e_{σ_f}		n_h	e_{n_h}		σ_h	e_{σ_h}	
		CV	no CV		CV	no CV		CV	no CV		CV	no CV
Estimating n												
CPV(90%)	9	4.75%	-0.08%	–	–	–	–	–	–	–	–	–
KPA	12	9.72%	-94.64%	–	–	–	–	–	–	–	–	–
RE	7	1.93%	-56.6%	–	–	–	–	–	–	–	–	–
5-fold CV	11	8.83%	-90.1%	–	–	–	8	0.64%	-28.9%	–	–	–
Estimating σ												
KTA	–	–	–	1.2	686%	607%	–	–	–	0.7	34%	7.51%
FSM	–	–	–	0.1	1776%	3129%	–	–	–	0.5	63.1%	12.5%
LVC	–	–	–	J_1:3	176%	154%	–	–	–	J_1:1.9	2.36%	1.12%
				J_2:2.6	216%	189%				J_2:2	2%	0.57%
MID	–	–	–	0.12	1779%	3107%	–	–	–	0.2354	166%	59.4%
DFN	–	–	–	2.3	253%	223%	–	–	–	3.4	1.04%	0.33%
GA	–	–	–	21.03	0.21%	-0.43%	–	–	–	4.11	9.47%	9.47%
MDM	–	–	–	33.06	0.38%	-0.58%	–	–	–	–	–	–
SVS	–	–	–	47.6	0.57%	-0.60%	–	–	–	–	–	–
SD	–	–	–	3.74	123%	110%	–	–	–	–	–	–
PVC	–	–	–	1200	0.75%	-0.61%	–	–	–	–	–	–
DaV	–	–	–	–	–	–	–	–	–	2.65($r=1$)	1.07%	-0.54%
										3.74($r=2$)	1.83%	9.09%
										5.92($r=5$)	5.36%	2.13%
										8.37($r=10$)	6.82%	3.27%
MCCV	–	–	–	–	–	–	–	–	–	3.1	0	-0.01%
SA	–	–	–	–	–	–	–	–	–	1.28	7.17%	2.46%
optimal CV framework	5			16.08			5			3.12		

Table 4: Estimation of n and σ for KPCA and KPLS models — mixing experiment.

Method	KPCA						KPLS					
	n_f	e_{n_f}		σ_f	e_{σ_f}		n_h	e_{n_h}		σ_h	e_{σ_h}	
		CV	no CV		CV	no CV		CV	no CV		CV	no CV
Estimating n												
CPV(90%)	8	5.68%	-14%	–	–	–	–	–	–	–	–	–
KPA	10	6.84%	-22.2%	–	–	–	–	–	–	–	–	–
RE	13	7.7%	-33.5%	–	–	–	–	–	–	–	–	–
5-fold CV	21	11.5%	-59%	–	–	–	16	19.1 %	-100%	–	–	–
Estimating σ												
KTA	–	–	–	15.2	32.7%	19.7%	–	–	–	15.2	701%	-30.6%
FSM	–	–	–	7.7	426%	76.4%	–	–	–	7.7	10120%	-100%
LVC	–	–	–	J_1:32	6.15%	8.03%	–	–	–	J_1:32	0.59%	0.73%
				J_2:30.1	6.64%	8.81%				J_2:30.1	0.39%	-7.9%
MID	–	–	–	0.00081	620%	579%	–	–	–	0.000813	13725%	-100%
DFN	–	–	–	34.6	5.77%	5.25%	–	–	–	27.8	6.55%	-37.9%
GA	–	–	–	127.89	2.28%	0.79%	–	–	–	17.14	345%	47.5%
MDM	–	–	–	4540	0.01%	-0.05%	–	–	–	–	–	–
SVS	–	–	–	5180	0.01%	-0.05%	–	–	–	–	–	–
SD	–	–	–	525.54	0.01%	-0.02%	–	–	–	–	–	–
PVC	–	–	–	14780	0.01%	-0.05%	–	–	–	–	–	–
DaV	–	–	–	–	–	–	–	–	–	38.4($r=1$)	21%	21.8%
										54.3($r=2$)	99.1%	21%
										85.9($r=5$)	179%	43.5%
										121.5($r=10$)	208%	69.7%
MCCV	–	–	–	–	–	–	–	–	–	30.9	0	0
SA	–	–	–	–	–	–	–	–	–	0.79	10818%	-100%
optimal CV framework	5			715.2			5			30.9		

Table 5: Summary of application studies including running time.

Application	KPCA			KPLS		
	n_f	σ_f	time	n_h	σ_h	time
Glass melter process	5	32.14	121min	5	0.62	102min
Distillation process	5	16.08	147min	5	3.12	116min
Mixing experiment	5	715.2	4min 18sec	5	30.9	50sec